

Interactive Optimization With Weighted Hypervolume Based EMO Algorithms: Preliminary Experiments

Dimo Brockhoff, Youssef Hamadi, Souhila Kaci

► **To cite this version:**

Dimo Brockhoff, Youssef Hamadi, Souhila Kaci. Interactive Optimization With Weighted Hypervolume Based EMO Algorithms: Preliminary Experiments. [Research Report] RR-8103, INRIA. 2012. <hal-00741730v2>

HAL Id: hal-00741730

<https://hal.inria.fr/hal-00741730v2>

Submitted on 12 Nov 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Interactive Optimization With Weighted Hypervolume Based EMO Algorithms: Preliminary Experiments

Dimo Brockhoff, Youssef Hamadi, Souhila Kaci

**RESEARCH
REPORT**

N° 8103

November 2012

Project-Team DOLPHIN



Interactive Optimization With Weighted Hypervolume Based EMO Algorithms: Preliminary Experiments

Dimo Brockhoff*, Youssef Hamadi†, Souhila Kaci‡§

Project-Team DOLPHIN

Research Report n° 8103 — November 2012 — 14 pages

* INRIA Lille - Nord Europe, DOLPHIN team, 59650 Villeneuve d'Ascq, France, dimo.brockhoff@inria.fr

† Microsoft Research, Cambridge, United Kingdom, youssefh@microsoft.com

‡ Université Montpellier 2, LIRMM, UMR 5506 - CC477, 161 rue Ada, 34095 Montpellier Cedex 5, France, kaci@lirmm.fr

§ In addition, all authors have been participating in the CNRS-Microsoft chair “Optimization for Sustainable Development (OSD)” at LIX, École Polytechnique, Palaiseau, France.

**RESEARCH CENTRE
LILLE – NORD EUROPE**

Parc scientifique de la Haute-Borne
40 avenue Halley - Bât A - Park Plaza
59650 Villeneuve d'Ascq

Abstract: The objective functions in multiobjective optimization problems are often non-linear, noisy, or not available in a closed form and evolutionary multiobjective optimization (EMO) algorithms have been shown to be well applicable in this case. Nowadays, for example within the scope of sustainable development, *many* objectives are taken into account: besides “classical” objectives such as cost and profit, some “new” objectives like energy consumption, noise levels or risks have to be considered.

With more and more objectives, the number of incomparable alternatives typically increases and the complexity of these problems does not make it easy for a decision maker to formalize preferences towards a specific solution or not even towards a specific but small enough portion of the search space. Moreover, also the algorithms themselves have difficulties to find a good approximation of the entire Pareto front if the number of incomparable solutions increases and the Pareto dominance relation does not indicate a good search direction anymore. In this case, combining the decision making with the search algorithm to an interactive optimization algorithm is considered as a valuable approach. While better and better solutions are found by the optimization algorithm, the DM can specify the preferences more and more precisely while learning about the problem and the objectives’ inherent tradeoffs. Such an interactive approach should profit from evaluating solutions only within the interesting regions of the search space in terms of a faster convergence towards the DM’s preferred solutions.

In the field of EMO, interactive optimization has only been considered recently and in comparison to the vast amount of general EMO algorithms, significantly less interactive EMO algorithms exist. Although, for example, optimization algorithms based on the weighted hypervolume indicator allow to incorporate various preference types into the search, no effort has been made to use this concept within an interactive algorithm. In this report, we propose and discuss how to combine interactive decision making and weighted hypervolume based search algorithms. We focus on a basic model where the DM is asked to pick the most desirable solution among a set. Several examples on standard test problems show the working principles and the usefulness of the interactive approach, in particular with respect to the proximity of the algorithm’s population to the DM’s most preferred solution.

Key-words: Multiobjective optimization, interactive decision making, evolutionary multiobjective optimization

Résumé : Les fonctions objectif en optimisation multi-objectif sont souvent non-linéaires, bruitées ou non-disponibles et l’optimisation multi-objectif évolutionnaire est applicable dans ce cas. De nos jours, par exemple dans le développement durable, *plusieurs* objectifs peuvent être pris en compte : en plus des objectifs “classiques” comme le coût et le profit, de “nouveaux” objectifs comme consommation d’énergie, niveaux de bruits ou de risque sont considérés.

Avec de plus en plus d’objectifs à prendre en compte, le nombre d’alternatives incomparables croît exponentiellement et la complexité de ces problèmes ne permet pas aux décideurs de formaliser ses préférences afin de calculer une solution spécifique ou même restreindre la recherche à un petit ensemble d’alternatives. De plus, les algorithmes ont des difficultés à trouver une bonne approximation de la région Pareto si le nombre d’alternatives incomparables est grand et la relation de dominance de Pareto ne permet plus une bonne direction de la recherche. Dans ce cas, combiner les algorithmes de recherche et la prise de décision en un algorithme d’optimisation interactif est considérée comme une approche alternative. Pendant que de meilleures solutions sont trouvées par l’algorithme d’optimisation, le décideur peut spécifier ses préférences de manière de plus en plus spécifique en apprenant le problème et le compromis entre les objectifs. Une telle approche interactive devrait bénéficier de l’évaluation des solutions seulement dans des régions intéressantes de l’espace de recherche en terme d’une convergence plus rapide vers les solutions préférées pour le décideur.

Dans le domaine de l’optimisation multi-objectif évolutionnaire, l’optimisation interactive a été seulement considérée récemment et en comparaison au grand nombre algorithmes d’optimisation multi-objectif évolutionnaire, peu d’algorithmes d’optimisation multi-objectif évolutionnaire interactifs existent. Bien que, par exemple, des algorithmes d’optimisation basés sur l’indicateur d’hyper-volume pondéré permettent d’inclure plusieurs types de préférences dans la recherche, aucun effort n’a été fourni pour utiliser ce concept dans les algorithmes interactifs. Dans ce rapport, nous proposons et discutons comment combiner la prise de décision interactive et les algorithmes de recherche basés sur l’hyper-volume pondéré. Nous considérons le modèle basique où le décideur est appelé à choisir les solutions qu’il préfère dans un ensemble de solutions. Plusieurs exemples de problèmes de tests standards montrent les principes et l’intérêt de l’approche interactive, en particulier par rapport à la proximité de la population de l’algorithme aux solutions préférées du décideur.

Mots-clés : optimisation multi-objectif, prise de décision interactive, optimisation multi-objectif évolutionnaire

1 Introduction

Multiobjective optimization problems with non-linear objectives which, in addition, can be noisy or not even given in a closed form occur frequently in practical applications. Evolutionary Multiobjective Optimization (EMO) algorithms have been shown to be applicable in such cases and are able to provide a decision maker (DM) with a set of solutions showing the tradeoffs among the objectives in a single algorithm run [10, 9]. However, problems often have many objectives and large search spaces where the current EMO algorithms need many function evaluations to converge to a good approximation. On the contrary, a DM is most of the time not interested in finding solutions *everywhere* at the Pareto front but only within certain *preferred regions* of it.

In such a scenario, it makes sense to combine the search for a solution set with the articulation of preferences by a DM in an *interactive* EMO algorithm in order to save function evaluations by exploring only the search space regions which are interesting to the DM [18, 11]. Several interactive EMO algorithms have been proposed in previous years but the amount of research on interactive EMO remains low when compared to the vast amount of general EMO algorithms [21, 18, 24, 16, 13, 19, 22]. Moreover, several of those studies assume a single preference model and a change in the preference modeling would need a different algorithm [21, 12, 24]. An approach which is able to integrate several preference models into the search is the weighted hypervolume indicator approach [25]. Its main idea is to define a weight function on the objective space and use the contribution to the weighted hypervolume indicator as the *fitness* of each solution in the environmental selection step of an EMO algorithm. Although it has been shown that several classical preference models such as weighted Tchebycheff utilities or desirability functions can be simulated with the weighted hypervolume approach in a set-based manner [2], it has not been used yet in an interactive fashion. It is the main goal of this paper to show how this can be achieved.

To this end, we assume a very basic scenario: In each interaction step, the DM has to decide on the most preferred solution within the EMO algorithm’s current population (or a subset thereof). The weight function of the weighted hypervolume indicator is then adapted accordingly such that more solutions close to the most-preferred one are found until the next interaction step. In particular, the paper

- briefly recapitulates the weighted hypervolume indicator and how it is employed in weighted hypervolume based search algorithms (Sec. 3),
- presents the basic framework in which the preference towards a specific solution in the algorithm’s population can be transformed into a weight function for the hypervolume indicator (Sec. 4), and
- shows in experiments on several test functions how the DM’s interactive choices affect the search (Sec. 5).

2 Preference Articulation And Interactive Optimization In Evolutionary Multiobjective Optimization

Classical EMO approaches aim at finding an approximation of the Pareto front while the DM decides *a posteriori* which solution in the computed set is the most preferred one [10, 9]. Recently proposed *interactive* EMO algorithms, on the other hand, involve the DM already during the search [18]—similar to their interactive counterparts from the classical multiple criteria decision making field [20]. At certain stages of the optimization that we will call “interaction steps” or

“interactive sessions”, the DM provides some kind of preference information which is then utilized by the EMO algorithm to find an approximation of the Pareto front which is biased towards the DM’s most preferred solutions. The known interactive EMO algorithms thereby differ mainly in the way the DM’s preference is modeled and used during the search.

According to [18], to which we refer for a broader overview of the topic, “probably the first interactive multiobjective metaheuristic” has been proposed by [23] as early as in 1993. The integration of the DM’s preference into “a relatively simple version of a Pareto-ranking based multi-objective evolutionary algorithm” [18] is very basic and implemented by duplicating solutions in the population that are desired by the DM and by removing undesired ones. More recent studies rely on more elaborate EMO algorithms than the 1993 genetic algorithm of [23] and allow for various other ways of preference articulation. Thiele et al. [24], for example, ask the DM to define desired solutions in objective space (so-called reference points) and an achievement scalarizing function [20] towards the current reference point is integrated into a binary quality indicator. This quality indicator is then, in turn, employed in the standard environmental selection operator of the indicator-based evolutionary algorithm (IBEA, [27]). Deb and Kumar [12] incorporate reference *directions* into the NSGA-II algorithm while Deb et al. [13] ask the DM to compare single solutions based on which a polynomial value function is created and optimized within the same NSGA-II algorithm. [21] ask the DM to compare solution pairs and integrate this information into the EMO algorithm. Fowler et al. [16] construct a partial order on the solutions by asking the DM to specify the best and worst solutions out of a specifically selected subset of the current EMO population whereas Deb et al. [13] directly ask for a full ranking of the presented solutions from the DM. Köksalan and Karahan [19] build their iTDEA algorithm around the interaction of selecting only the best solution within the presented subset of the population. This specification of the most preferred solution among a set of solutions is also the scenario, we build upon in the following.

In [1], several ways to articulate the DM’s preferences within the class of *weighted hypervolume based EMO algorithms* are presented. Together with the weighted hypervolume’s property of being compliant with the Pareto dominance relation [28], it has been shown both theoretically and practically that this approach allows to steer the search towards preferred solutions or regions of the objective space [1, 2]. In this study, we show that the weighted hypervolume algorithm W-HypE [1, 2] can also be used in an interactive fashion. To this end, the information about the most preferred solutions specified by the DM is used to define a weight function that has larger values around the preferred solutions—resulting in populations which accumulate close to the solutions most-preferred in the previous interaction step. We will see later how exactly the proposed interactive W-HypE algorithm can focus the search and how close the found solutions are to the solutions most-preferred by the DM.

3 The Weighted Hypervolume Indicator and Hypervolume-Based Selection

Throughout the paper, we assume minimization of k objective functions f_i mapping a solution $x \in X$ from the search space X to a real value. The vector-valued objective function $f(x) = (f_1(x), \dots, f_k(x))$ maps a solution from the search space X to its objective vector $f(x)$ in the so-called objective space \mathbb{R}^k . We call a solution x^* *Pareto-optimal* if there is no other solution $x \in X$ such that x^* is *dominated* by x or, more formally, if there is no other $x \in X$ such that $\forall 1 \leq i \leq k : f(x) \leq f(x^*)$ and $\exists 1 \leq i \leq k : f(x) < f(x^*)$. The set of all Pareto-optimal solutions is called Pareto set and its image in objective space Pareto front. In its most general form, the weighted hypervolume indicator is then a set quality measure which assigns a set (or multi-set)

of solutions $A \subseteq X$ with objective vectors $\{f(a) \mid a \in A\} \subseteq \mathbb{R}^k$ a real number: the weighted Lebesgue measure of the objective space dominated by solutions in A , bounded by a reference point $r \in \mathbb{R}^k$ and weighted by $w : \mathbb{R}^k \rightarrow \mathbb{R}$ [25]:

$$I_{H,w}(A, r) = \int_{\mathbb{R}^k} w(z) \mathbf{1}_{H(A,r)}(z) dz \quad (1)$$

where $H(A, r) = \{z \in \mathbb{R}^k \mid \exists a \in A : f(a) \leq z \leq r\}$ and $\mathbf{1}_S$ is the indicator function of a set S , i.e., $\mathbf{1}_S(s) = 1$ if $s \in S$ and $\mathbf{1}_S(s) = 0$ otherwise. In case of $w(z) = 1$ for all $z \in \mathbb{R}^k$, we use the term (standard) hypervolume indicator (I_H).

The (weighted) hypervolume indicator is used frequently for performance assessment of multiobjective optimizers [29] but also in several recent EMO algorithms as optimization criterion within their environmental selection [4, 17, 3]. One of the main reasons for its popularity is the fact that the (weighted) hypervolume indicator is compliant with the dominance relation [28]. This compliance implies that only Pareto-optimal solutions are found if the (weighted) hypervolume indicator is optimized [15].

However, optimizing the hypervolume indicator exactly within the environmental selection step of an EMO algorithm is often not possible due to the complexity of the problem. If, for example, λ solutions are newly generated within each iteration but the number of stored solutions is restricted to μ , the optimal strategy would be to keep the μ points among all possible sets of μ points that maximizes the hypervolume indicator. However, this computation is typically exponential in μ as often $\mu = \lambda$ is chosen. Hence, a greedy strategy is performed in all known hypervolume based EMO algorithms. This sometimes also *1-greedy strategy* called approach [5] deletes the solution with the worst *hypervolume contribution* iteratively until only μ solutions remain where the hypervolume contribution of a solution $a \in A$ with respect to A is defined as $I_H(A) - I_H(A \setminus \{a\})$. In the case of a steady-state algorithm where in each iteration only one new solution is generated [4] and therefore also only one solution has to be deleted, the 1-greedy strategy is optimal. If, however, more than one solution has to be deleted in each generation, the 1-greedy strategy is known to compute sub-optimal solution sets, the hypervolume of which can be arbitrarily bad [8]. In such a situation, the *expected hypervolume loss* of a solution $a \in X$ if a itself and $i - 1$ other randomly chosen solutions are deleted can be used as the quality of a . This is the idea behind the algorithm HypE of [3] which is also to be used in this study. If the weighted hypervolume indicator is employed within the algorithm, we denote this variant as W-HypE. As the hypervolume computation itself is already computationally expensive [7], W-HypE uses Monte Carlo sampling to estimate the weighted hypervolume indicator in Equation 1. Throughout the paper, we use 10,000 samples in each iteration of the algorithm. For further details about the algorithm, we refer to [3] and [1].

4 Interactive Optimization With Weighted Hypervolume-Based Selection

4.1 Basic Concept

The basic idea behind the proposed approach is to ask the DM about the most preferred solutions among the current population of the EMO algorithm every fixed number of G generations. Once the most preferred solutions are known, they are used as the means of Gaussian weight functions within the W-HypE algorithm while the direction of the distributions are determined by the extreme points of the current population. Together with the selection scheme of W-HypE, this

will drive the population towards regions with larger weight function, thus, towards the solutions preferred by the DM.

As Gaussian weight functions, which are furthermore efficient to sample, we use the ones proposed in [1]. Let $P = \{x_1, \dots, x_{|P|}\}$ be the current population and $b \in P$ the best solution picked by the DM. Then, $m = f(b)$ shall be the mean of the chosen Gaussian distribution and t its direction such that the new weight function equals

$$w(z) = \frac{1}{(2\pi)^{k/2}|C|^{1/2}} e^{-\frac{1}{2}(z-m)^T C^{-1}(z-m)} \quad (2)$$

where the covariance matrix $C := \sigma_\varepsilon^2 \mathbf{I} + \sigma_t^2 t t^T / \|t\|^2$ has eigenvectors t, t_2, \dots, t_k with eigenvalues $\sigma_\varepsilon^2 + \sigma_t^2, \sigma_\varepsilon^2, \dots, \sigma_\varepsilon^2$ while t_2, \dots, t_k are taken from an orthogonal basis of the hyperplane orthogonal to t . The determinant of C is thereby denoted as $|C|$. At first sight, σ_t and σ_ε are two additional parameters of the approach. However, it seems to be clear that they both should depend on the distribution of points which is the reason why we propose to choose them proportional to the population's spread.

In the current implementation, two overlapping Gaussians are used to incorporate the information about a preferred point b where the first one is sampled with 80%, the second with 20% of the samples. If not stated otherwise, the direction is $t = (1, 1)$, the variance along this direction is chosen as $\sigma_t = 0.5 \cdot \ell$, and the second variance is chosen as $\sigma_\varepsilon = 0.1 \cdot \ell$ (for the first Gaussian) and as $\sigma_\varepsilon = 0.01 \cdot \ell$ for the second where ℓ is the Euclidean distance between the current nadir and ideal points, i.e., $\ell = \|(f_{1,\max} - f_{1,\min}, f_{2,\max} - f_{2,\min})\|$ with $f_{i,\max}$ the maximal and $f_{i,\min}$ the minimal value in objective i obtained for solutions in the current population. If more than one solution are preferred equally by the DM, two Gaussians are defined for each of them and the number of samples are distributed equally among the preferred points.

Until the first interaction step with the DM, the interactive W-HypE algorithm is using the standard hypervolume indicator in order to come up with a good spread of the solutions before the DM's decisions change the weight function as described above¹.

4.2 A First Example on How to Steer the Search Interactively

In order to show that the above approach is working, we implemented the interactive W-HypE algorithm within the algorithm package PISA [6]². Two example test runs are shown here in which the bi-objective DTLZ2 and DTLZ7 test problems [14] are optimized. The population size is set to $\mu = \lambda = 20$ and the number of decision variables is 100. All other parameters have been set to their standard choices, given by the PISA implementation. The interaction with the DM took place every $G = 100$ iterations of the algorithm.

In the first example, shown in the left plot of Fig. 1, the DM decided that the solution closest to $f_1 = 0.9$ is the most desired solution in each step. In the second example, shown in the right plot of Fig. 1, we simulated a highly indecisive DM who changed the preferences from the leftmost to the rightmost solution of the current population in every interaction step. We show this result in order to check whether the interactive W-HypE algorithm can also cope with drastically changing preferences.

What can be seen from the examples of Fig. 1 is that the interactive W-HypE algorithm follows the directions specified by the DM, i.e., the population is moving towards the selected solution as well as towards better solutions in terms of Pareto dominance. Note that the shown examples stem from single algorithm runs which, due to the stochasticity of the algorithm, do not

¹Note that, as the current implementation is using Monte Carlo sampling, also the standard hypervolume in the first iterations of the algorithm is sampled here.

²The source code will be made available online at <http://inrialix.gforge.inria.fr/interactive/>.

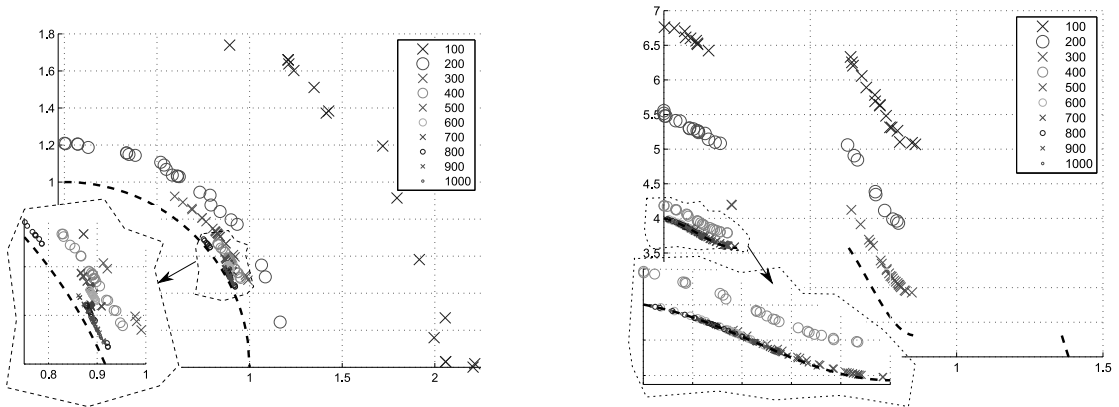


Figure 1: Examples of interactive optimization runs on the DTLZ2 test function when the DM is preferring the solution closest to $f_1 = 0.9$ (left) and on the DTLZ7 test function when the DM is alternatingly preferring the leftmost and then the rightmost solution of the population (right).

give statistically sound results in general. However, additional runs show the same tendencies in the resulting populations (not shown). In the following, we will investigate the parameter settings of the approach in more detail and obtain statistically sound results with independent algorithm runs.

5 On Setting the Parameters of the Indicator’s Weight Functions

5.1 Different Directions of the Gaussian Weight Vectors

In the above example, the Gaussian weight functions have been chosen to be aligned with the diagonal $(1, 1)$. However, this choice of the Gaussian’s direction t is arbitrary and other choices might give better results, depending on the problem at hand and moreover depending on where the preferred solution lies. If the DM, for example, selects one of the outmost points of the current population, the choice of the direction should result in a similar concentration of solutions around the preference point than if a solution in the center of the current non-dominated front is selected.

Here, we investigate exemplary three specific ways to determine the direction t : (i) independent of the chosen solution as above ($t = (1, 1)$), (ii) starting from the preference point p towards the current ideal point, or (iii), connecting the nadir point and p , see Fig. 2. Figure 3 shows the results of some example runs on the DTLZ2 and DTLZ7 functions where the DM chose alternatingly the leftmost and the rightmost solution in the current population as the preferred point. All parameters except the direction t of the Gaussian weight function are kept constant as in the above examples.

Results: For both test problems, the different directions of the Gaussians show a general characteristic³. When using the direction towards the nadir point (case iii above), the population is losing less diversity when compared to using the fixed direction $t = (1, 1)$ (case i). When using the direction towards the ideal point (case ii), the diversity of the population becomes

³Note that although the shown results stem from single runs, the results are representative and can be observed for other runs as well.

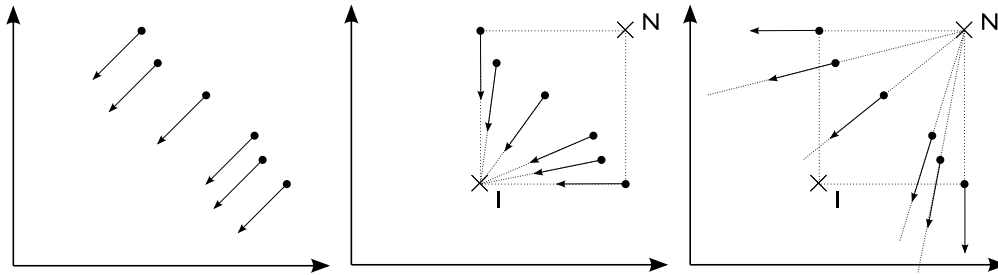


Figure 2: Three possibilities to choose the direction t within the Gaussian weight function approach: independent of the chosen preference point always along the diagonal (left), in direction to the ideal point I (middle), or in direction to the nadir point N (right).

even smaller than in case i. In particular when the DM prefers a solution at the extremes of the current population, and even more if the preferred extreme changes with each interaction as in the examples here, keeping the diversity of the population high seems to be a good strategy. Hence, we will always use case iii in the following.

5.2 Different Shapes of the Gaussian Weight Functions

Another parameter of the approach is the shape of the Gaussian weight functions, in particular the values of σ_t and σ_ε . Here, we compare several algorithm variants where the value of σ_ε is changing while σ_t is kept constant. In order to make statistically sound statements about the different parameter settings, we run every algorithm variant 10 times and simulate the DM in each interaction step. To this end, we assume that the (simulated) DM is deciding on the most preferred solution by choosing the solution with the highest weighted Tchebycheff utility function $u(s) = -\max_{j \in 1 \dots k} \lambda_j |z_j^* - z_j|$ for a given weight $\lambda = (\lambda_1, \dots, \lambda_k)$ and $z^* = (z_1^*, \dots, z_k^*)$ being a predefined point, not worse than any feasible objective vector (here $z^* = 0^k$). In the following, we use $\lambda = (0.4, 0.6)$ for the bi-objective ZDT1 problem [26]. In this case, the optimal point, giving the highest utility over all feasible solutions, is $p_{\text{opt}} = (x_{\text{opt}}, 1 - \sqrt{x_{\text{opt}}})$ with $x_{\text{opt}} = \frac{21}{8} - \frac{3}{8}\sqrt{33} \approx 0.470789$ which can be computed as the intersection point between the line $l(x) = \frac{2}{3}x$, given by the weighted Tchebycheff function, and the line $f(x) = 1 - \sqrt{x}$ describing the test function's Pareto front.

Figure 4 shows boxplots of the minimum and mean Euclidean distance (in objective space) between this optimal point and the solutions found by HypE and several interactive W-HypE variants with two Gaussian weight functions and variances $\sigma_{\varepsilon,1} = 10 \cdot \sigma \cdot \ell$ and $\sigma_{\varepsilon,2} = \sigma \cdot \ell$ ($\sigma \in \{1, 10^{-1}, 10^{-2}, 10^{-3}\}$). The value ℓ is thereby the Euclidean distance between the current nadir and ideal points and $\sigma_t = 0.5 \cdot \ell$. The interaction took place every 100 generations.

Results: For both the minimum and mean distances, W-HypE with the parameter value of $\sigma = 0.1$ is clearly the best algorithm whereas for the other W-HypE variants, no clear ranking can be given. In any case, the interaction steps in W-HypE together with the focus of the weight function to the preferred objective space region improves the distances over the standard HypE algorithm with a fixed hypervolume indicator.

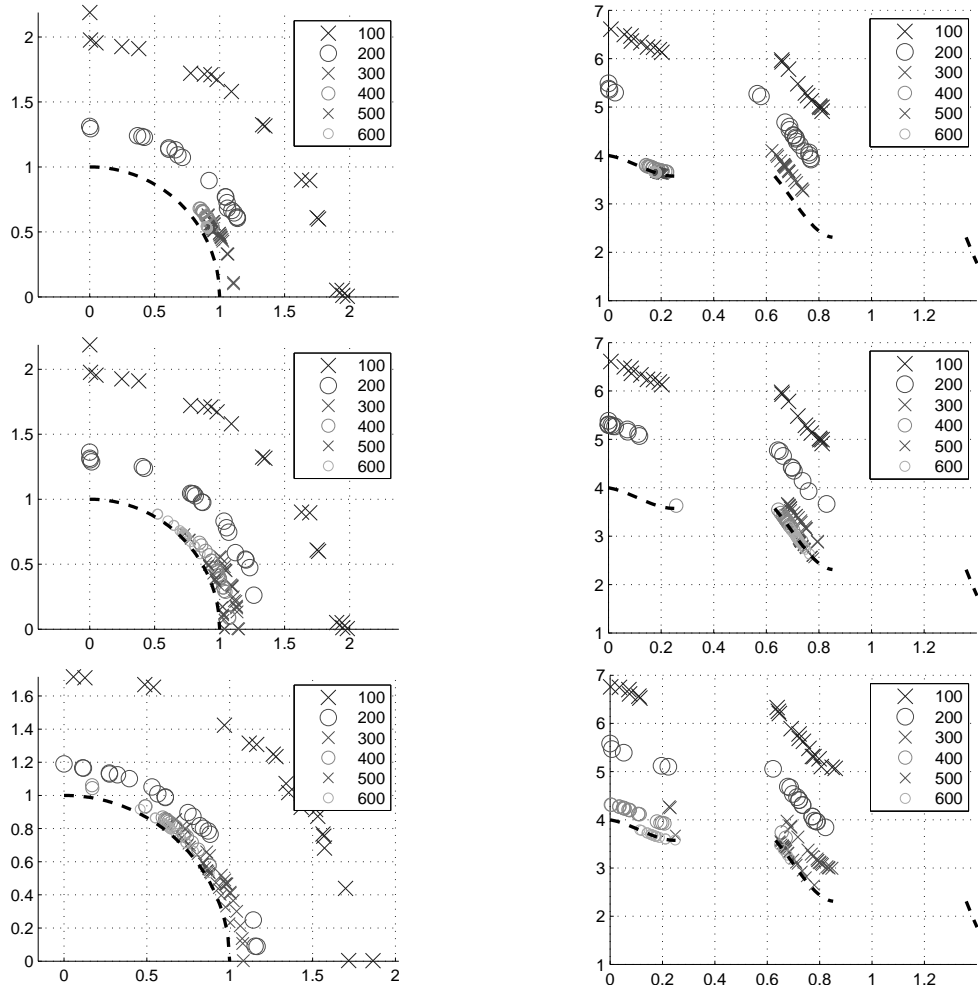


Figure 3: Exemplary runs on DTLZ2 (left) and DTLZ7 (right) problems with different directions of the Gaussian weight function while the DM chooses the preferred solution alternatively as the leftmost and rightmost solution: diagonal direction (upper row), direction from point to ideal point (middle), and direction from nadir point to preferred solution (third row).

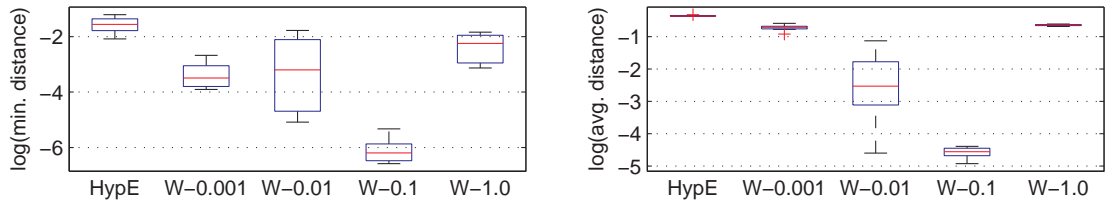


Figure 4: Boxplots showing the minimal (left) and mean (right) distances of the HypE and the interactive W-HypE populations after 1000 generations on the DTLZ2 problem to the optimal point of the weighted Tchebycheff function with weights (0.4, 0.6) that has been used to simulate the DM of the interactive W-HypE. The label W- σ stands for the W-HypE version with two Gaussian weight functions and variances $\sigma_{\varepsilon,1} = 10 \cdot \sigma \cdot \ell$ and $\sigma_{\varepsilon,2} = \sigma \cdot \ell$ with ℓ the Euclidean distance between the current nadir and ideal points.

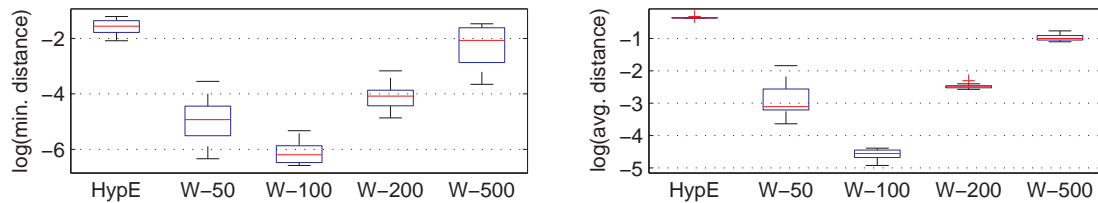


Figure 5: Boxplots showing the minimal (left) and mean (right) distances of the HypE and the interactive W-HypE populations after 1000 generations to the optimal point of the weighted Tchebycheff function with weights $(0.4, 0.6)$ that has been used to simulate the DM of the interactive W-HypE. The label W- G stands for the W-HypE version with an interactive decision making step every G generations.

5.3 Changing the Number of Decision Maker Calls

Even in case the number of generations of the algorithm is kept constant, the number of interactive steps where the DM specifies the preferences is a parameter of the algorithm and it is important to investigate its influence. To this end, we run 10 independent runs of W-HypE for each of the following parameter settings on the bi-objective ZDT1 problem: $G = 50$, $G = 100$, $G = 200$, and $G = 500$ which, with the number of generations chosen as 1000 as above, results in 20, 10, 5, and 2 DM calls. We choose to simulate the DM again with an underlying weighted Tchebycheff with weights $(0.4, 0.6)$. Figure 5 shows the corresponding boxplots of the distances to the optimal Pareto-optimal point for the weighted Tchebycheff function.

Results: We see that the tendencies for the average and minimal distances are the same and that the value of $G = 100$ produces the best results among all tested choices for G on this problem. It is therefore clear that too few as well as too many DM calls result in sub-optimal results when compared with respect to the distance to the optimal solution. We therefore recommend to use around 10 DM calls for the currently used parameter settings and the problems investigated in this study. Additional investigations, however, need to be done in the future.

6 Conclusions

Interactive Evolutionary Multiobjective Optimization (EMO) gained recent interest in the research community—combining the set-based randomized EMO algorithms with interactive decision maker (DM) sessions in which the DM defines preferences towards solutions of interests. These preferences are then employed in the EMO algorithm to steer the search towards preferred parts of the search space [18, 11]. Optimizing the so-called weighted hypervolume indicator in a hypervolume based EMO algorithm has been shown to allow to steer the search [25, 1, 2]. It is therefore straightforward to use this approach also in an interactive manner. In this work, we presented a simple way to incorporate the information about the DM’s most preferred solutions into the weighted hypervolume based W-HypE algorithm by means of multivariate normal distributions. We also showed its working principles and the influence of its main parameters on the obtained solution qualities on several standard test problems, in particular with respect to the proximity of the algorithm’s population to the DM’s most preferred solution.

References

- [1] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Articulating User Preferences in Many-Objective Problems by Sampling the Weighted Hypervolume. In G. Raidl et al., editors, *Genetic and Evolutionary Computation Conference (GECCO 2009)*, pages 555–562, New York, NY, USA, 2009. ACM.
- [2] J. Bader, D. Brockhoff, L. Thiele, and E. Zitzler. Directed Multiobjective Optimization Based on the Hypervolume Indicator. *Journal of Multi-Criteria Decision Analysis*, 2013. accepted for publication.
- [3] J. Bader and E. Zitzler. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. *Evolutionary Computation*, 19(1):45–76, 2011.
- [4] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective Selection Based on Dominated Hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.
- [5] N. Beume, B. Naujoks, M. Preuss, G. Rudolph, and T. Wagner. Effects of 1-Greedy S -Metric-Selection on Innumerably Large Pareto Fronts. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2009)*, pages 21–35. Springer, 2009.
- [6] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler. PISA—A Platform and Programming Language Independent Interface for Search Algorithms. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, pages 494–508. Springer, 2003.
- [7] K. Bringmann and T. Friedrich. Approximating the Volume of Unions and Intersections of High-Dimensional Geometric Objects. In *International Symposium on Algorithms and Computation (ISAAC 2008)*, pages 436–447. Springer, 2008.
- [8] K. Bringmann and T. Friedrich. Don't Be Greedy When Calculating Hypervolume Contributions. In *Foundations of Genetic Algorithms (FOGA 2009)*, pages 103–112. ACM, 2009.
- [9] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, Berlin, Germany, 2007.
- [10] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, Chichester, UK, 2001.
- [11] K. Deb and M. Köksalan. Guest Editorial: Special Issue on Preference-based Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 14:669–670, 2010.
- [12] K. Deb and A. Kumar. Interactive Evolutionary Multi-Objective Optimization and Decision-Making using Reference Direction Method. In *Genetic and Evolutionary Computation Conference (GECCO 2007)*, pages 781–788. ACM, 2007.
- [13] K. Deb, A. Sinha, P. Korhonen, and J. Wallenius. An Interactive Evolutionary Multi-Objective Optimization Method Based on Progressively Approximated Value Functions. *IEEE Transactions on Evolutionary Computation*, 14(5):723–739, 2010.

- [14] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. In A. Abraham, R. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, chapter 6, pages 105–145. Springer, 2005.
- [15] M. Fleischer. The Measure of Pareto Optima. Applications to Multi-Objective Metaheuristics. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, pages 519–533. Springer, 2003.
- [16] J. W. Fowler, E. S. Gel, M. Köksalan, P. Korhonen, J. L. Marquis, and J. Wallenius. Interactive Evolutionary Multi-Objective Optimization for Quasi-Concave Preference Functions. *European Journal of Operational Research*, 206(2):417–425, 2009.
- [17] C. Igel, N. Hansen, and S. Roth. Covariance Matrix Adaptation for Multi-objective Optimization. *Evolutionary Computation*, 15(1):1–28, 2007.
- [18] A. Jaszkiewicz and J. Branke. Interactive Multiobjective Evolutionary Algorithms. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 179–193. Springer, 2008.
- [19] M. Köksalan and I. Karahan. An Interactive Territory Defining Evolutionary Algorithm: iTDEA. *IEEE Transactions on Evolutionary Computation*, 14(5):702–722, 2010.
- [20] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer, Boston, MA, USA, 1999.
- [21] S. Phelps and M. Köksalan. An Interactive Evolutionary Metaheuristic for Multiobjective Combinatorial Optimization. *Management Science*, 49(12):1726–1738, 2003.
- [22] A. Sinha, K. Deb, P. Korhonen, and J. Wallenius. An Interactive Evolutionary Multi-Objective Optimization Method Based on Polyhedral Cones. In *Learning and Intelligent Optimization (LION 2010)*, pages 318–332, 2010.
- [23] T. Tanino, M. Tanaka, and C. Hojo. An Interactive Multicriteria Decision Making Method by Using a Genetic Algorithm. In *Conference on Systems Science and Systems Engineering*, pages 381–386, 1993.
- [24] L. Thiele, K. Miettinen, P. K. Korhonen, and J. Molina. A Preference-Based Interactive Evolutionary Algorithm for Multiobjective Optimization. *Evolutionary Computation*, 17(3):411–436, 2009.
- [25] E. Zitzler, D. Brockhoff, and L. Thiele. The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, pages 862–876. Springer, 2007.
- [26] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, 2000.
- [27] E. Zitzler and S. Künzli. Indicator-Based Selection in Multiobjective Search. In X. Yao et al., editors, *Conference on Parallel Problem Solving from Nature (PPSN VIII)*, volume 3242 of *LNCS*, pages 832–842. Springer, 2004.
- [28] E. Zitzler, L. Thiele, and J. Bader. On Set-Based Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 14(1):58–79, 2010.

- [29] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.



**RESEARCH CENTRE
LILLE – NORD EUROPE**

Parc scientifique de la Haute-Borne
40 avenue Halley - Bât A - Park Plaza
59650 Villeneuve d'Ascq

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399